Deep & Shallow Relation Extraction with Probabilistic Inference

Malcolm Greaves

greaves.malcolm@gmail.com

https://github.com/malcolmgreaves/talks/blob/master/deep_shallow_re_prob.pdf

What does this presentation cover?

Presentation Topics

- Showcase of different approaches to relation extraction
  - Shallow linguistic feature classifiers
  - Deep end-to-end models
  - Probabilistic inference (graphical models & logic)
- Mixing in work I completed in my master’s thesis at CMU
- Presentation of where new methods can go in, integrate, and upgrade this line of work

What is a relation?

Relations: A Primer

- A relation is a property that exists between **entities**
  - e.g. “People have professional titles”
    - e.g. “A person may be a US State Senator”
- People write **evidence** for relations in text:
  - “In 2017, Kamala D. Harris was sworn in as a United States Senator for California, the second African-American woman and first South Asian-American senator in history.”

https://www.harris.senate.gov/about

What is relation extraction?

Relation Extraction

"In 2017, Kamala D. Harris *was sworn in* as a United States Senator *for California*, the *second African-American woman* and *first South Asian-American senator in history*."

→

```
per_title(
    Kamala_Harris,
    US_Senator
)
```

Relation Extraction

"In 2017, Kamala D. Harris *was sworn in* as a United States Senator for *California*, the *second African-American woman* and *first South Asian-American senator in history*."

Unstructured Text

```
per_title(
  Kamala_Harris,
  U.S._Senator
)
```

Structured Knowledge

Outline for Rest of Presentation

Outline

- ~~Relation Primer~~
- Background on relation extraction literature
- Data labeling
- Shallow, multi-sentence RE w/ Probabilistic 1st order logic
- Deep Relation Extraction: Graph LSTMs
- Building a Knowledge Graph: PSL for Collective Inference
- Meta patterns in Relation Extraction

Brief Background on
Relation Extraction Literature

Essential Questions for RE Systems

- Are my relations pre-determined?
  - ∃ schema of typed relations vs. Open IE
- How do I know if this text contains a relation?
  - Heuristic over-generate & filter? End-to-end task?
- Do I extract relations between *mentions* or *entities*?
- Where’s my labeled data?
- Extraction scope: Sentence? Paragraph? Document? Corpus?

Relation Classification Models: Broadly

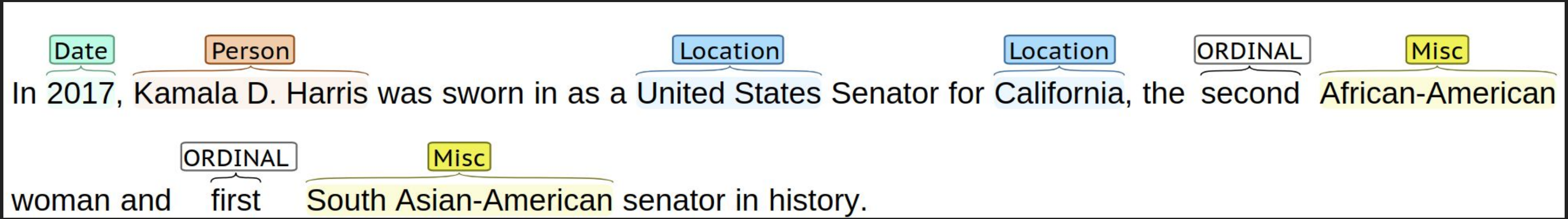
- Heuristics
  - e.g. Hearst Patterns (lexico-syntactic patterns)
    - e.g. "__________ **X**'s spouse, **Y**, __________."
    - "Automatic Acquisition of Hyponyms from Large Text Corpora." Hearst (1992)
- Shallow feature classifiers
- Deep neural networks
- Graphical models
- Probabilistic logic
- Open IE

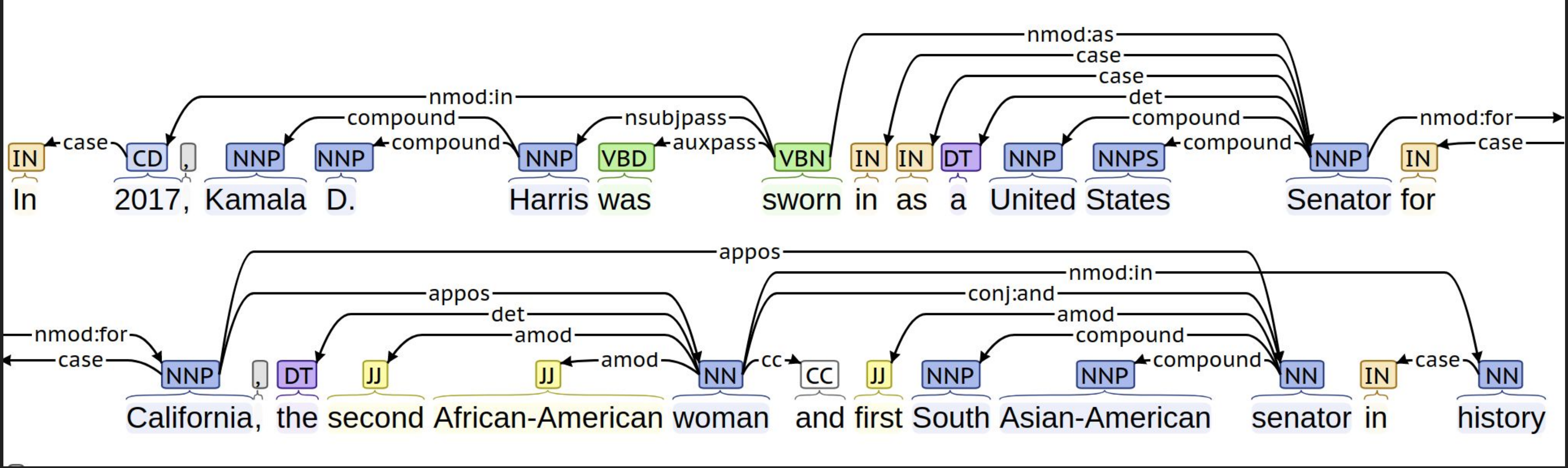
Relation Classification Models: **Covered**

- Heuristics
  - e.g. Hearst Patterns (lexico-syntactic patterns)
    - e.g. "_________ **X**'s spouse, **Y**, _________."
      - "Automatic Acquisition of Hyponyms from Large Text Corpora." Hearst (1992)
- **Shallow feature classifiers**
- **Deep neural networks**
- Graphical models
- **Probabilistic logic**
- Open IE

Shallow Relation Extraction

- Linguistic or syntactic features
  - e.g. ngrams, dependency paths, POS tags
- Linear models for large-scale problems
  - e.g. logistic regression, hinge-loss linear SVM
  - >1e7 features: L1/2 regularization helps
- Non-linear SVMs & random forests
- Fixed labeled data for model training
- Often heuristic over-generate & filter for real-world use





[“The Stanford CoreNLP Natural Language Processing Toolkit.” Manning, et. al. (2014)]

Publications: Shallow RE

- “Open information extraction for the web.” Banko et. al (2007)
- “Toward an architecture for never-ending language learning” Carlson et. al. (2010)
- “Large-scale learning of relation extraction rules with distant supervision from the web.” Krause, Li, Uszkoreit, and Xu (2012)
- “Relation Extraction using Distant Supervision, SVMs, and Probabilistic First Order Logic.” Greaves (2014)
- “Effective Slot Filling Based on Shallow Distant Supervision Methods.” Roth, Barth, Weigand, Singh, and Klakow (2014)
- “Exploiting Shallow Linguistic Information for Relation Extraction from Biomedical Literature.” Giuliano, Lavelli, and Romano (2006)

Deep Relation Extraction

- Recurrent neural networks (RNNs):
  - LSTMs & GRUs: naturally “read” through text
  - Context retains long-distance information
- Convolutional Neural Networks
  - local windows convolve over word embeddings
- No feature engineering & good with word sparsity
- End-to-end task learning (entity identification, relation prediction); current State-of-the-Art task performance

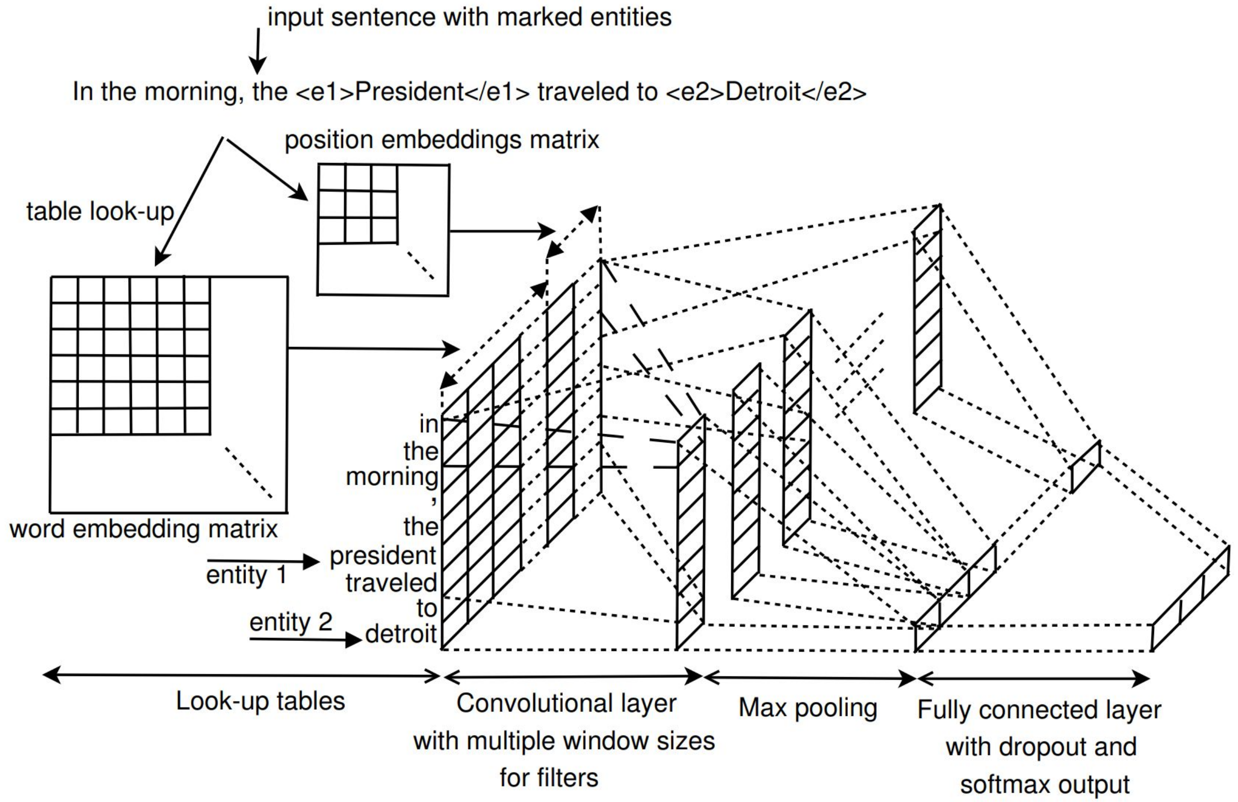


Figure 1: Convolutional Neural Network for Relation Extraction.

[“Relation Extraction: Perspective from Convolutional Neural Networks.” Nguyen and Grishman (2015)]

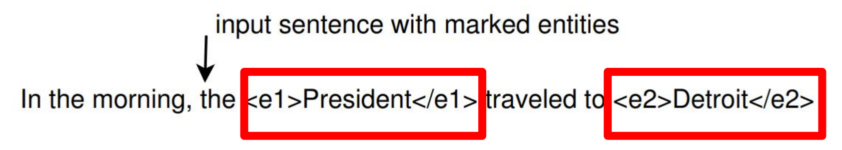


Figure 1: Convolutional Neural Network for Relation Extraction.

[“Relation Extraction: Perspective from Convolutional Neural Networks.” Nguyen and Grishman (2015)]

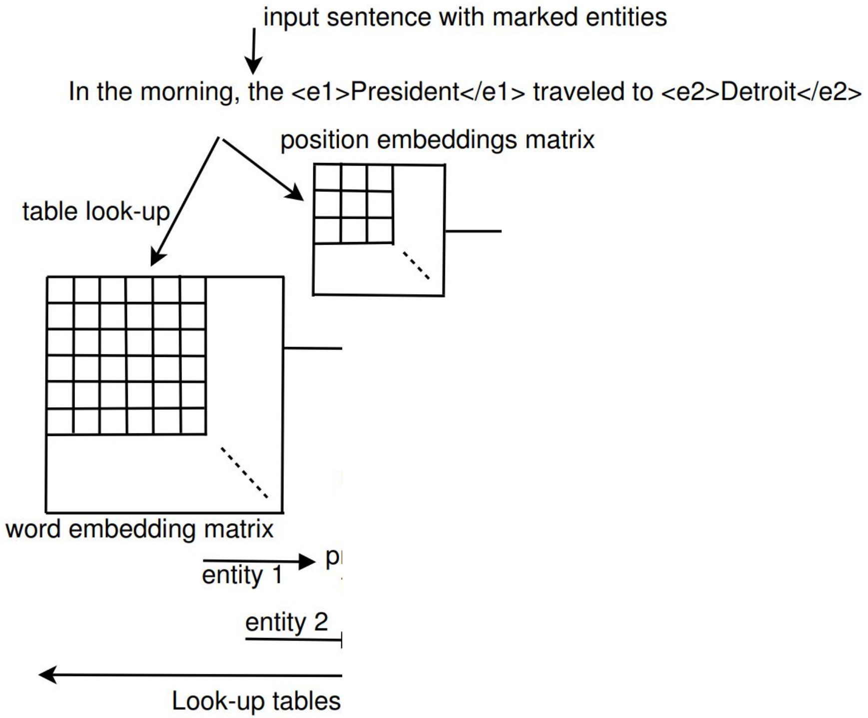


Figure 1: Convolutional Neural Network for Relation Extraction.

[“Relation Extraction: Perspective from Convolutional Neural Networks.” Nguyen and Grishman (2015)]

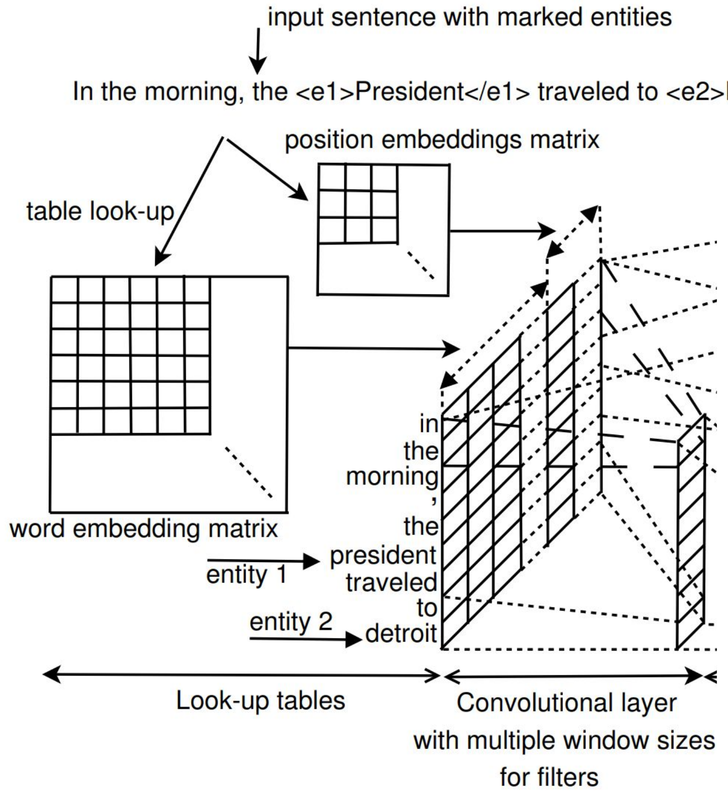


Figure 1: Convolutional Neural Network for Relation Extraction.

[“Relation Extraction: Perspective from Convolutional Neural Networks.” Nguyen and Grishman (2015)]()

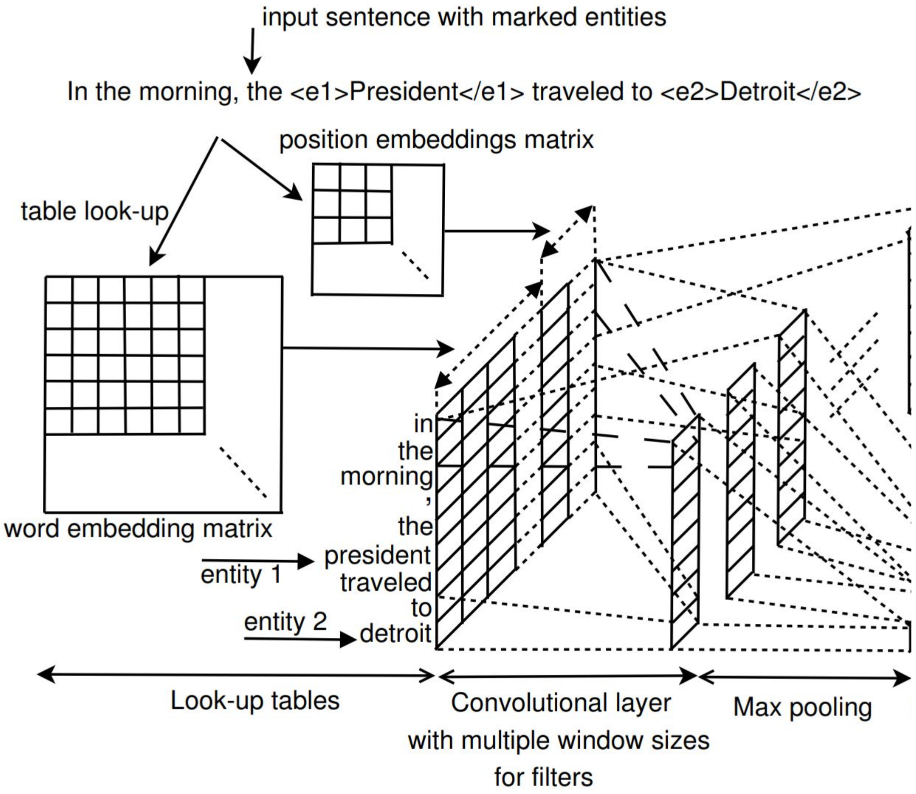


Figure 1: Convolutional Neural Network for Relation Extraction.

[“Relation Extraction: Perspective from Convolutional Neural Networks.” Nguyen and Grishman (2015)]

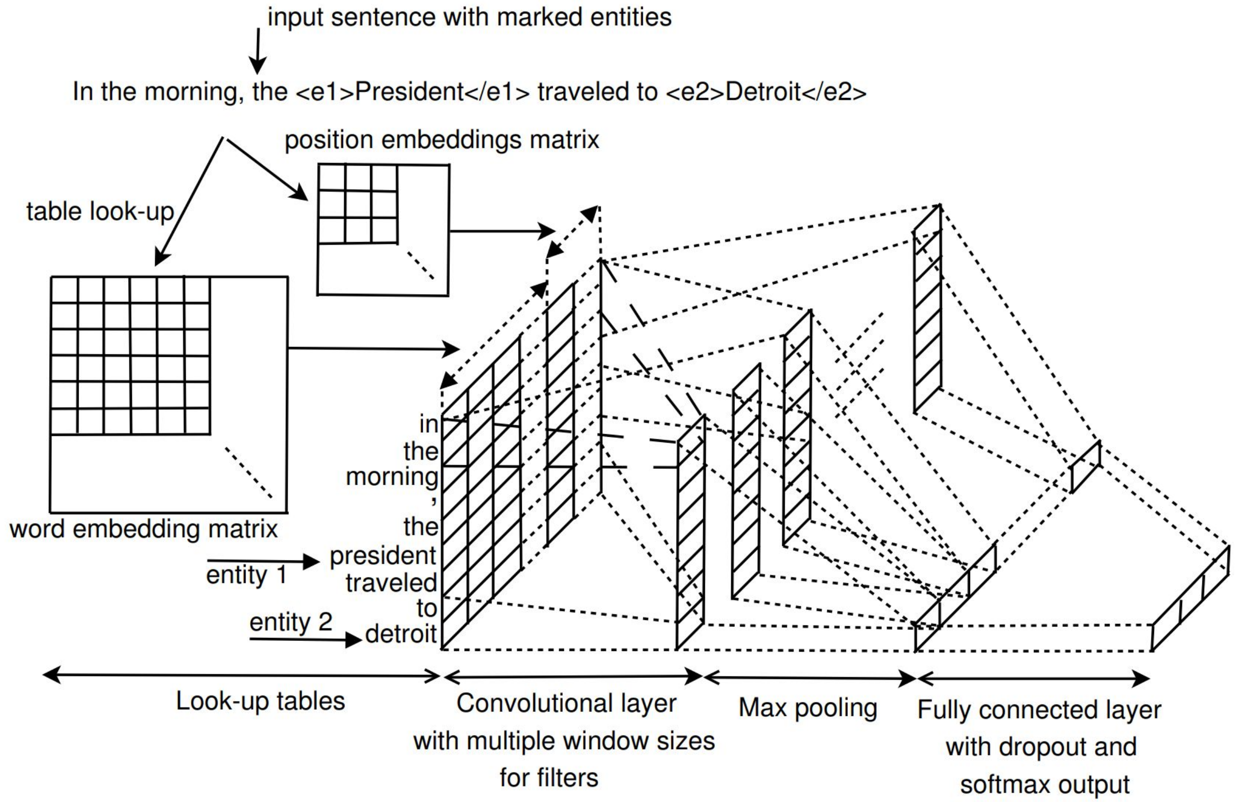


Figure 1: Convolutional Neural Network for Relation Extraction.

"Relation Extraction: Perspective from Convolutional Neural Networks." Nguyen and Grishman (2015)

Publications: Deep RE

- "Cross-Sentence N-ary Relation Extraction with Graph LSTMs." Peng, Poon, Quirk, Toutanova, and Yih (2017)
- "Snorkel MeTaL: Weak Supervision for Multi-Task Learning" Ratner, Hancock, Dunnmon, Goldman, and Ré (2018)
  - * Hybrid system: human-generated weak data labeling functions & deep model training
- "Relation Extraction: Perspective from Convolutional Neural Networks." Nguyen and Grishman (2015)
- "Relation Extraction with Multi-instance Multi-label Convolutional Neural Networks." Jiang, Wang, Li, and Wang (2016)
- "Incorporating Relation Paths in Neural Relation Extraction." Zeng, Lin, Liu, and Sun (2017)

Graphical Models for Relation Extraction

-

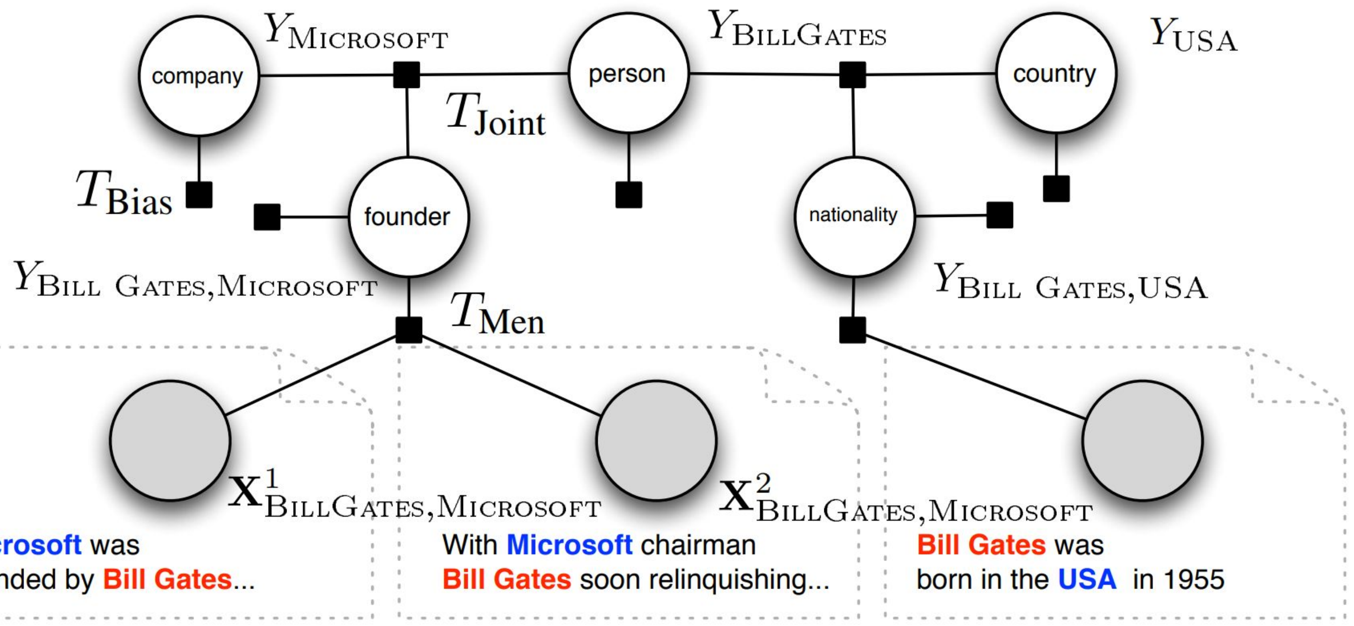


Figure 1: Factor Graph of our model that captures selectional preferences and functionality constraints. For readability we only label a subsets of equivalent variables and factors. Note that the graph shows an example assignment to variables.

"Collective Cross-Document Relation Extraction Without Labelled Data." Yao, Riedel, and McCallum (2005)

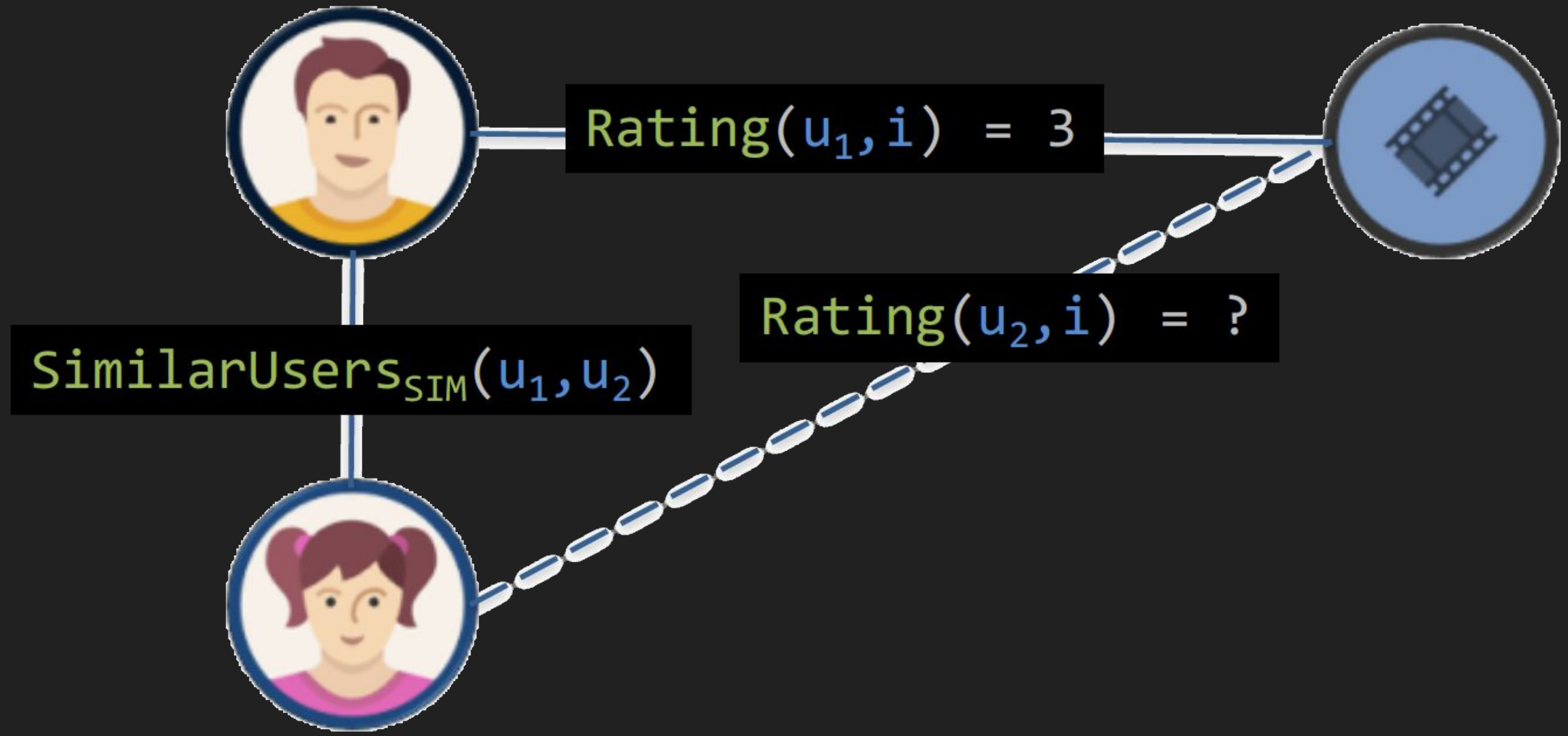
Publications: Graphical Models for RE

- "Collective Cross-Document Relation Extraction Without Labelled Data." Yao, Riedel, and McCallum (2005)
- "FACTORIE: Efficient Probabilistic Programming for Relational Factor Graphs via Imperative Declarations of Structure, Inference and Learning" (McCallum, Rohanemanesh, Wick, Schultz, and Singh (2008)
- "Represent, Aggregate, and Constrain: A Novel Architecture for Machine Reading from Noisy Sources." Naradowsky & Riedel (2017)
- "A Survey of Distant Supervision Methods using PGMs." Madden (2017)

Probabilistic Logic for RE

- Entities exist as variables
- Logical rules relate variable relationships
  - e.g. `subclass(X,Y)` for `X=`cookie, `Y=`baked goods
- Input data is noisy
  - Multiple, redundant or possibly conflicting evidence
- Inference system finds maximally consistent set
  - Rules (relations) are *grounded* with variables (entities)
  - Each grounding has truth value $\in [0,1]$

$\text{SimilarUsers}_{SIM}(u_1,u_2)$ & $\text{Rating}(u_1,i)$ -> $\text{Rating}(u_2,i)$



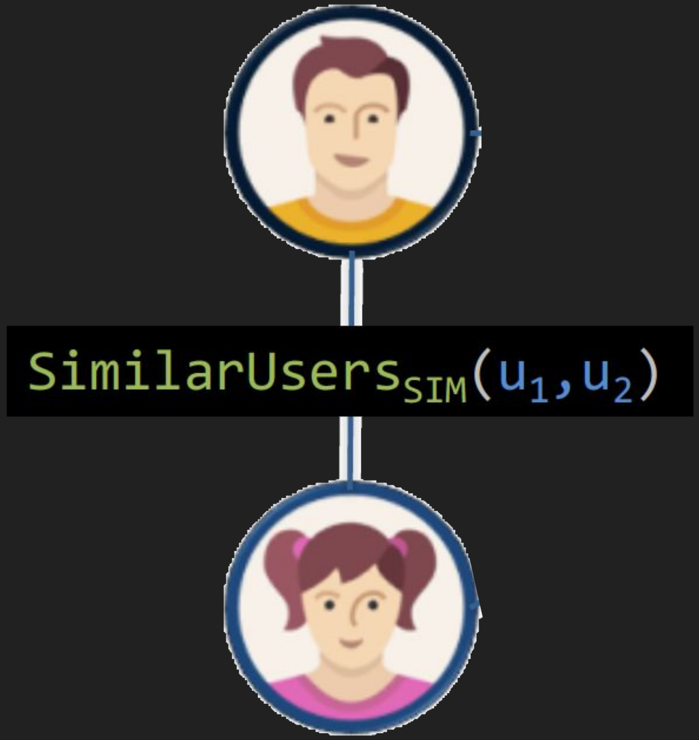
“HyPER: A Flexible and Extensible Probabilistic Framework for Hybrid Recommender Systems.” Kouki et. al. 2015

```
SimilarUsers_SIM(u_1,u_2) & Rating(u_1,i) -> Rating(u_2,i)
```

$$\mathrm{SimilarUsers}_{SIM}(u_1,u_2)\ \&\ \mathrm{Rating}(u_1,i)\ \text{->}\ \mathrm{Rating}(u_2,i)$$

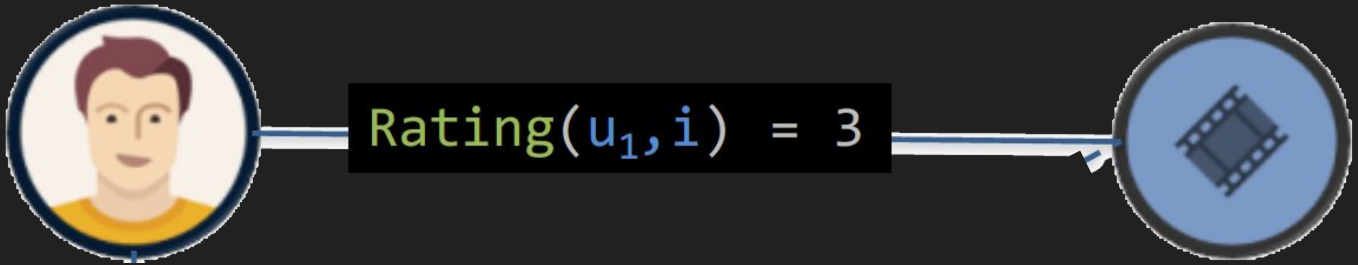
“HyPER: A Flexible and Extensible Probabilistic Framework for Hybrid Recommender Systems.” Kouki et. al. 2015

$\mathrm{SimilarUsers}_{SIM}(u_1,u_2)$ & $\mathrm{Rating}(u_1,i)$ -> $\mathrm{Rating}(u_2,i)$



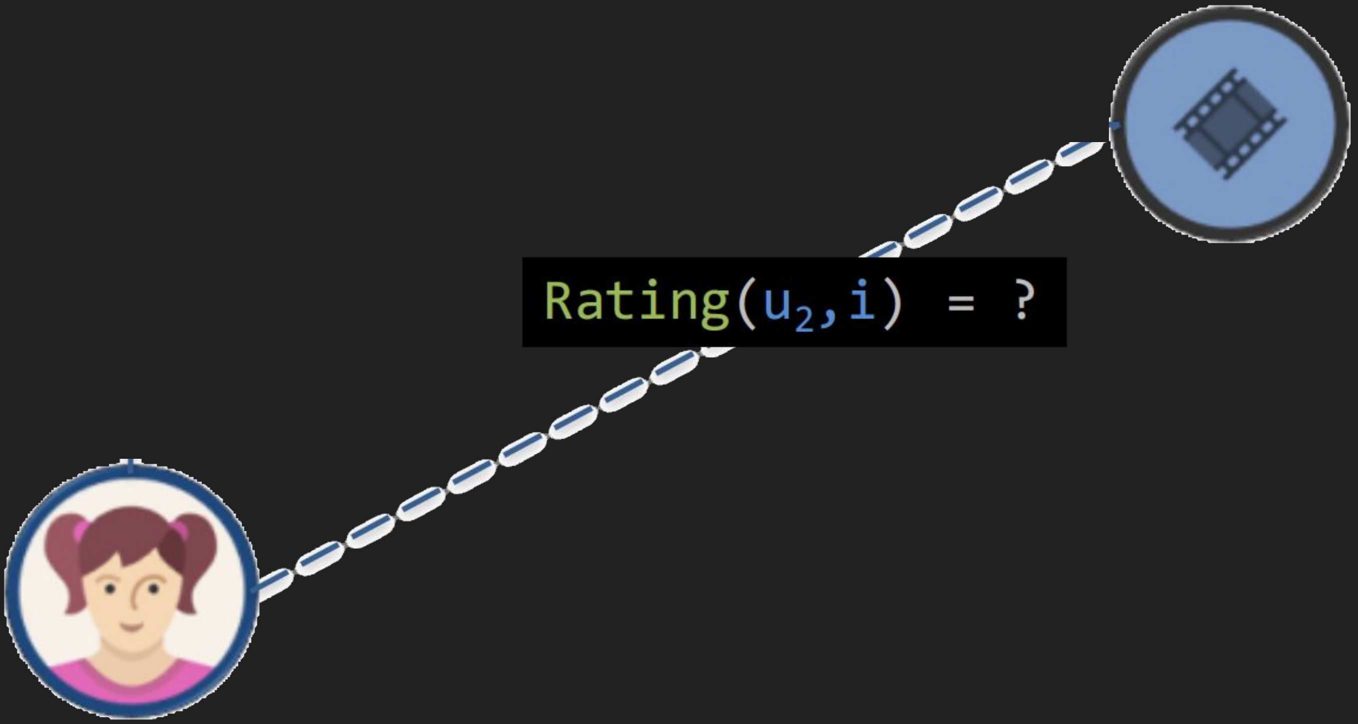
[“HyPER: A Flexible and Extensible Probabilistic Framework for Hybrid Recommender Systems.” Kouki et. al. 2015]

$\mathrm{SimilarUsers}_{\mathrm{SIM}}(u_1,u_2)\ \&\ \mathrm{Rating}(u_1,i)\ \text{->}\ \mathrm{Rating}(u_2,i)$



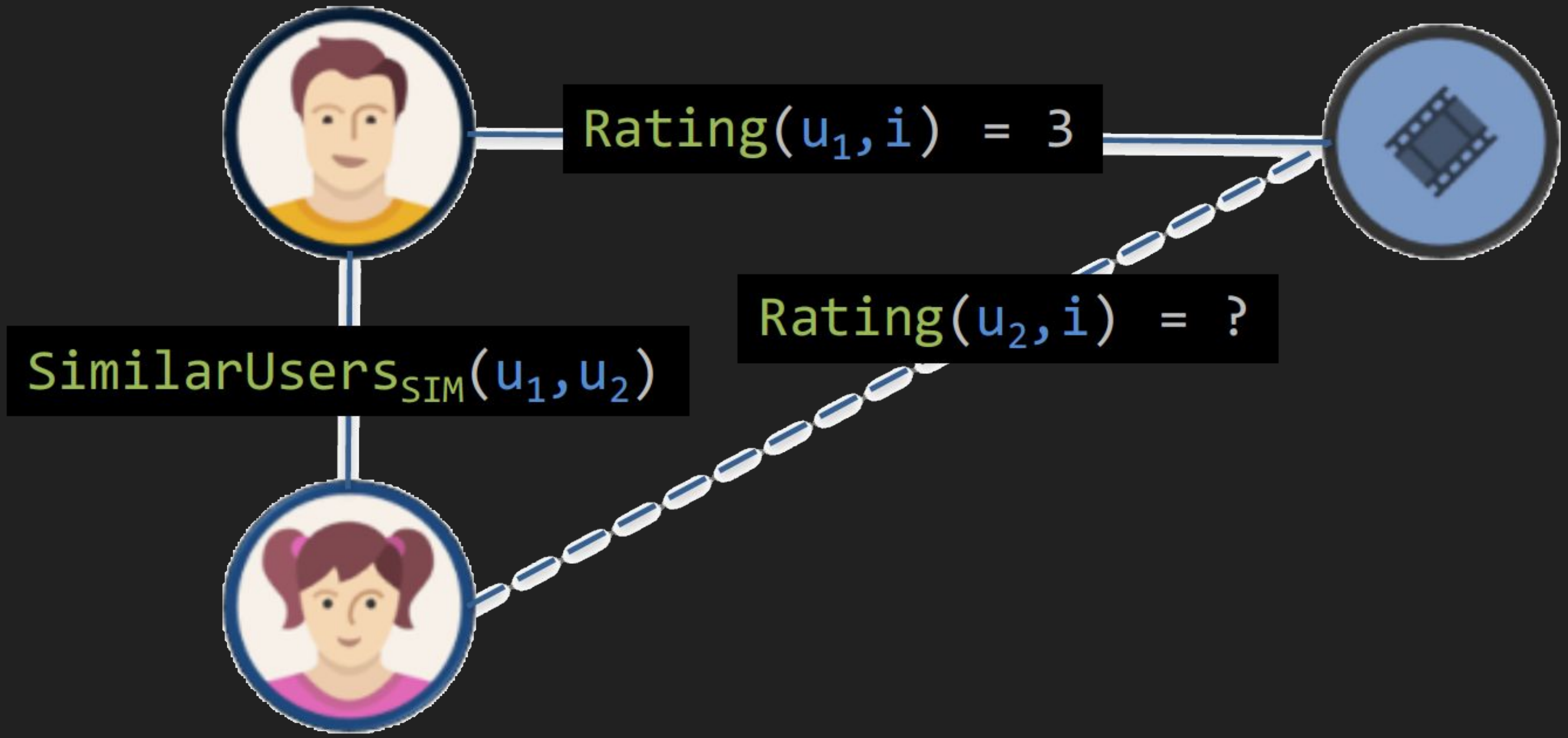
"HyPER: A Flexible and Extensible Probabilistic Framework for Hybrid Recommender Systems." Kouki et. al. 2015

$$\mathrm{SimilarUsers}_{SIM}(u_1,u_2)\ \&\ \mathrm{Rating}(u_1,i)\ \rightarrow\ \mathrm{Rating}(u_2,i)$$



[“HyPER: A Flexible and Extensible Probabilistic Framework for Hybrid Recommender Systems.” Kouki et. al. 2015]

$\mathrm{SimilarUsers}_{SIM}(u_1,u_2)\ \&\ \mathrm{Rating}(u_1,i) \rightarrow \mathrm{Rating}(u_2,i)$



“HyPER: A Flexible and Extensible Probabilistic Framework for Hybrid Recommender Systems.” Kouki et. al. 2015

Publications: Probabilistic Logic RE

- "Probabilistic Declarative Information Extraction." Wang, Michelakis, Franklin, Garofalakis, and Hellerstein (2010)
- "Machine Reading Using Markov Logic Networks for Collective Probabilistic Inference." Ghosh, Shankar, and Owre (2011)
- "Markov Logic for Machine Reading." Poon (2011)
- "Online Inference for Relation Extraction with a Reduced Feature Set." Rabinovich and Archambeau (2015)
- "Large-Scale Knowledge Graph Identification Using PSL." Pujara, Miao, Getoor, and Cohen (2013)
- "Relational Markov Networks for Collective Information Extraction" Bunescu and Mooney (2004)

Labeled Data for Relation Extraction

Labeling Data for Relation Extraction

- Labeled data is necessary for **evaluation** & supervised learning
- Annotating text by hand is hard, time-consuming → **$$$$**
- Make noisy training data: let statistical model sort it out
  - Bootstrapping & iterate (e.g. NELL)
  - Distant supervision (using e.g. Freebase, DBPedia)
  - Labeling functions (e.g. Snorkel)

Bootstrap & Iterate

- Seed each relation from schema with few examples *(~50)*
- Train **high-precision** classifiers with seed examples
  - Learn feature importance
- Accept highest scoring / probable extractions @ iteration *k*
  - Add as positive examples for training iteration *k+1*
- Suffers from **semantic drift**
  - Training on an incorrectly classified example □
  - e.g. “Internet Cookie” is a “baked good”

Distant Labeling

- Have existing knowledge base of relation facts
- Have massive amounts of unstructured text
- **Assumption**: Every tuple of entities identified in text is a real-world example of all applicable relations.
  - ☻ "Barack Obama is married to Michelle Obama."
  - ☹ "Barack and Michelle were seen in town today."
- Noisily-labeled training data → let sheer volume of data and statistical model sort-through noise

Labeling Functions

- Deterministic function that labels an entity tuple in text with a **+** or **-** relation label
- Often simple heuristics that use linguistic or syntactic features for calculation
    - E.g. “At least one word from {husband, wife, spouse} occurs in the sentence → entity pair is example for ‘married’ relation”
- Noisily-labeled training data (similar to distant supervision)

Master’s Thesis Dive

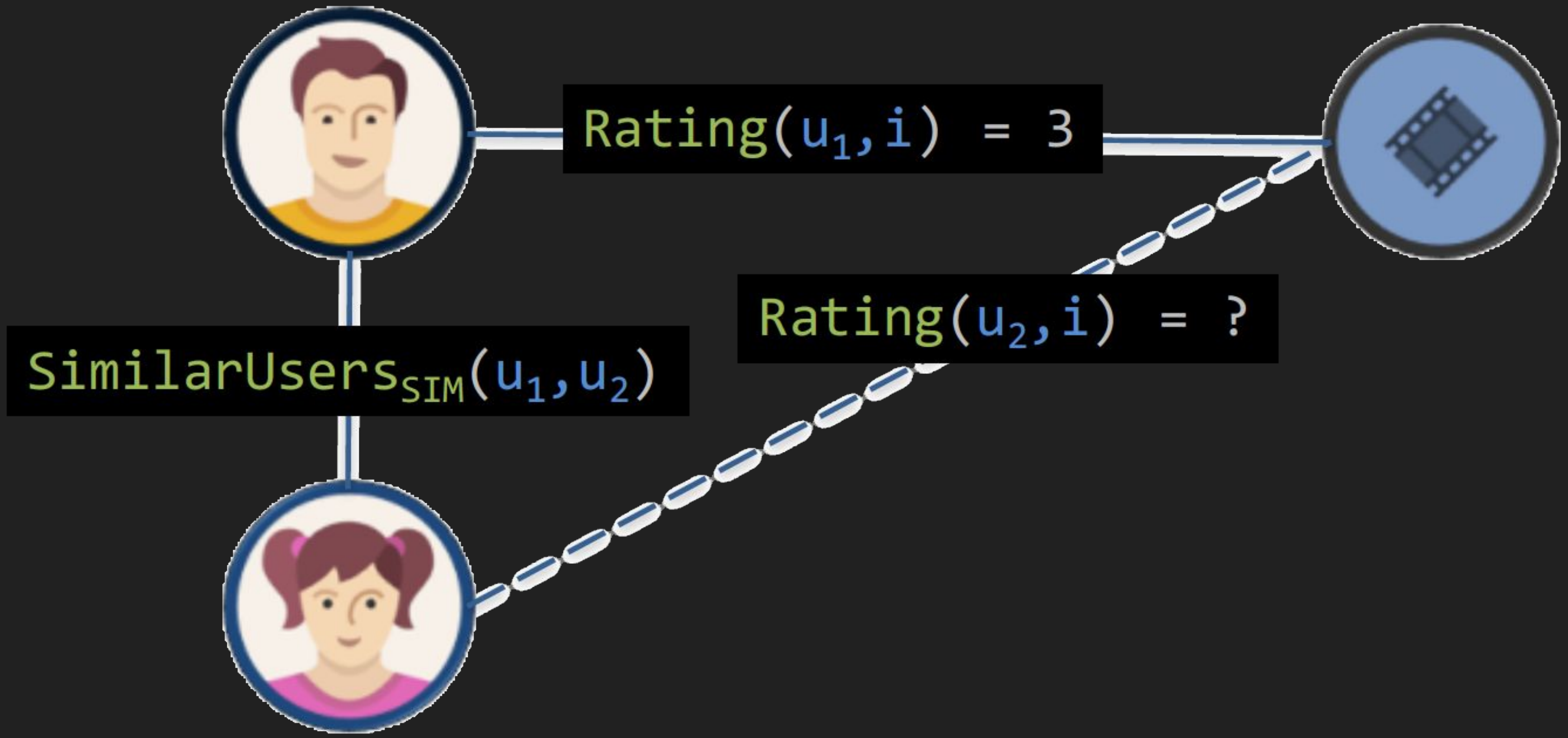
What does this system do?

- [“Relation Extraction using Distant Supervision, SVMs, and Probabilistic First Order Logic.” Greaves (2014)]
- [Re-written Code] https://github.com/malcolmgreaves/rex
- NLP (tokenize, POS tag, NER, coreference resolution)
- Distant supervision with NELL KB
- Over-generate candidates with ProPPR rules
- N-gram, skip n-gram features in linear SVMs
- Binary SVM per relation & a “no-relation” classifier

ProPPR for Candidate Generation

- We can construct rules to describe valid relation extraction *candidates*
  - Put them in 1st order logical form
- Identify all instances of rules in a text corpus
  - Build a proof graph for all entity mentions
- Use probabilistic inference to find a maximally consistent set of candidates
  - ProPPR uses approximate inference for scaling

$$\text{SimilarUsers}_{\text{SIM}}(u_1,u_2)\ \&\ \text{Rating}(u_1,i)\ \text{->}\ \text{Rating}(u_2,i)$$



[“HyPER: A Flexible and Extensible Probabilistic Framework for Hybrid Recommender Systems.” Kouki et. al. 2015]

Atomic Statements: Defined in Data

$\texttt{entInSent}(Q, S)$	$:=$	Entity Q is in sentence S
$\texttt{sentHasEnt}(S, A)$	$:=$	Sentence S has entity A
$\texttt{sentInDoc}(S, D)$	$:=$	Sentence S in in document D
$\texttt{docHasSent}(D,)$	$:=$	Document D has sentence S
$\texttt{sentHasEntPOS}(S, A, \texttt{T})$	$:=$	Entity A in sentence S has POS tag $\texttt{T}$
$\texttt{coref}(Ref, Sq, A, Sa)$	$:=$	Ref in sentence Sq is a referent to A in sentence Sa

Across the Sentence Boundary

$$\texttt{sentlink}(Q, S, A) \quad \texttt{:-} \quad \texttt{entInSent}(Q, S),\ \texttt{sentHasEnt}(S, A)\ .$$

Entities **Q** and **A** are in sentence **S**

$$\begin{aligned}\texttt{doclink}(Q, Sq, Sa, A) \quad \texttt{:-} \quad & \texttt{entInSent}(Q, Sq),\ \texttt{sentInDoc}(Sq, D) \\ & \texttt{docHasSent}(D, Sa),\ \texttt{sentHasEnt}(Sa, A)\ .\end{aligned}$$

Entity **Q** is in a sentence
Entity **A** is in another
Both sentences are in the same document **D**

Candidate Constraints

constraint(S, A) :- sentHasEntPOS(S, A, NN) .
constraint(S, A) :- sentHasEntPOS(S, A, NNS) .
constraint(S, A) :- sentHasEntPOS(S, A, NNP) .
constraint(S, A) :- sentHasEntPOS(S, A, NNPS) .

Entity **A** in sentence **S** must have a noun POS tag

Final Candidate Rules: Sentence vs. Doc

$$\texttt{candidateSent}(Q, S, A) \quad \texttt{:-} \quad \texttt{sentlink}(Q, S, A),\ \texttt{constraint}(S, A)\ .$$

Sentence candidate

$$\begin{aligned}\texttt{candidateDoc}(Q, Sq, Sa, Ref, A) \quad \texttt{:-} \quad & \texttt{doclink}(Q, Sq, Sa, Ref),\ \texttt{sentlink}(Ref, Sa, A),\\ & \texttt{coref}(Q, Sa, Ref, Sa),\ \texttt{constraint}(S, A)\ .\end{aligned}$$

Two-sentence candidate: `Q` has referent

$$\begin{aligned}\texttt{candidateDoc}(Q, Sq, Sa, Ref, A) \quad \texttt{:-} \quad & \texttt{sentlink}(Q, Sq, Ref),\ \texttt{doclink}(Ref, Sq, Sa, A),\\ & \texttt{coref}(Ref, Sq, A, Sa),\ \texttt{constraint}(S, A)\ .\end{aligned}$$

Two-sentence candidate: `A` has referent

Cross-Sentence Example

He was born in 1961 in Honolulu, Hawaii, two years after the territory was admitted to the Union as the 50th state.

Raised largely in Hawaii, Obama also spent one year of his childhood in Washington state and four years in Indonesia.

https://en.wikipedia.org/wiki/Barack_Obama

Entities, Anaphora, and Referent

He was born in 1961 in **Honolulu, Hawaii**, two years after the territory was admitted to the Union as the 50th state.

Raised largely in Hawaii, **Obama** also spent one year of his childhood in Washington state and four years in Indonesia.

https://en.wikipedia.org/wiki/Barack_Obama

Intuition: Turn into Single-Sentence!

Obama was born in 1961 in **Honolulu, Hawaii**, two years after the territory was admitted to the Union as the 50th state.

- Perform coreference resolution
- Substitute anaphora with referent
- Now treat as single-sentence relation extraction

N-gram feature SVM

- N-gram features surrounding entities
  - Left window: “A B C D <u>entity</u>” → (A,B), (B,C), (C, D)
  - Right: “<u>entity</u> E F G H” → (E,F), (F,G), (G,H)
- k-skip n-gram features between entities
  - N-grams with wild-cards → skipping (up to) `k` tokens
    - Skip `0, 1, 2, 3, …, k`
  - Captures long distance information, combats sparsity

Cost-Sensitive Learning

- Normal SVM objective function treats all errors as the same
- With imbalanced training data, leads to dreadful performance
    - Too many negatives? Abysmal False-Positive rate
        - Over-generate & filter here meant too many negatives
    - Too many positives? Abysmal False-Negative rate
- Fix? **Not all errors are the same!**

Cost-Sensitive Learning

$$\text{Minimize:} \quad \frac{1}{2}||\vec{w}||^2 + C \sum \xi_i$$

$$\text{subject to:} \quad \forall k : y_k[\vec{w} \cdot \vec{x_k} + b] \geq 1 - \xi_k$$

Cost-Sensitive Learning

$$\text{Minimize:} \quad \frac{1}{2}||\vec{w}||^2 + \boxed{C_+}\sum_{i:y_i=1}\xi_i + \boxed{C_-}\sum_{j:y_j=-1}\xi_j$$

$$\text{subject to:} \quad \forall k : y_k[\vec{w} \cdot \vec{x}_k + b] \geq 1 - \xi_k$$

Results on TAC-KBP 2014 Relations

(macro-averaged)	**Precision**	**Recall**	**F1**
Within-Sentence	34.45%	27.13%	29.29%
Cross-Sentence	36.35%	28.37%	29.54%
Full ProPPR Within-Sent. + X-Sent.	**38.66%**	**31.86%**	**33.15%**

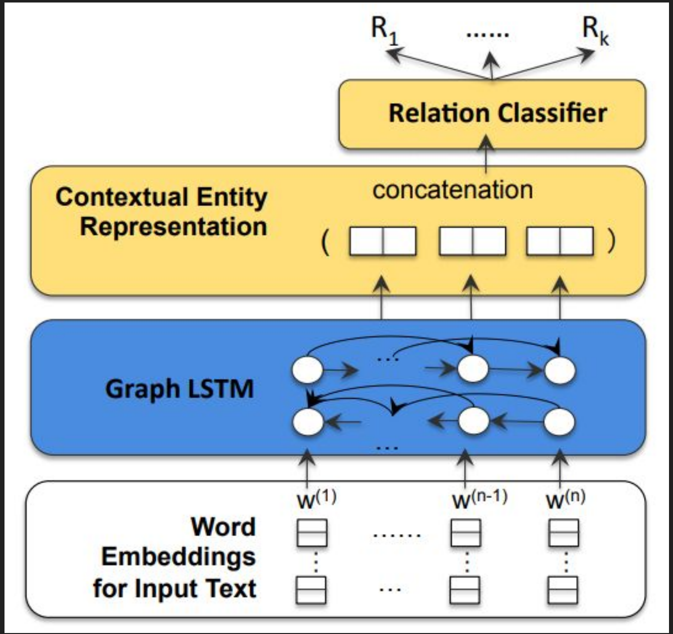
Graph LSTM Dive

What does this system do?

- “Cross-Sentence N-ary Relation Extraction with Graph LSTMs.” Peng, Poon, Quirk, Toutanova, and Yih (2017)
- N-ary relation extraction *(more than the binary examples!)*
- Bi-directional LSTM that learns entity tuple & relation embeddings
- Extracts from multiple sentences: document-level
- Incorporates syntactic dependency paths
- Multi-task learning: all relations learned simultaneously
- End-to-end entity ID & relation classification → extraction

Learning Relation Embeddings

- Input is word embedding sequence
- Bi-directional Graph LSTM layer builds context, hidden state embeddings
- Outputs embedding for entity tuple
- Concatenated vector fed into logistic regression + softmax for relation classification



"Cross-Sentence N-ary Relation Extraction with Graph LSTMs." Peng, Poon, Quirk, Toutanova, and Yih (2017)

Document Graph

- Nodes are words
- Edges:
  - syntactic dependencies (from dependency parse)
  - Linear context (adjacent words)
    - No dependency edges → normal LSTM
- If sentence `A` precedes sentence `B`, then the root of `A` has a directed edge to the root of `B`.

Document Graph

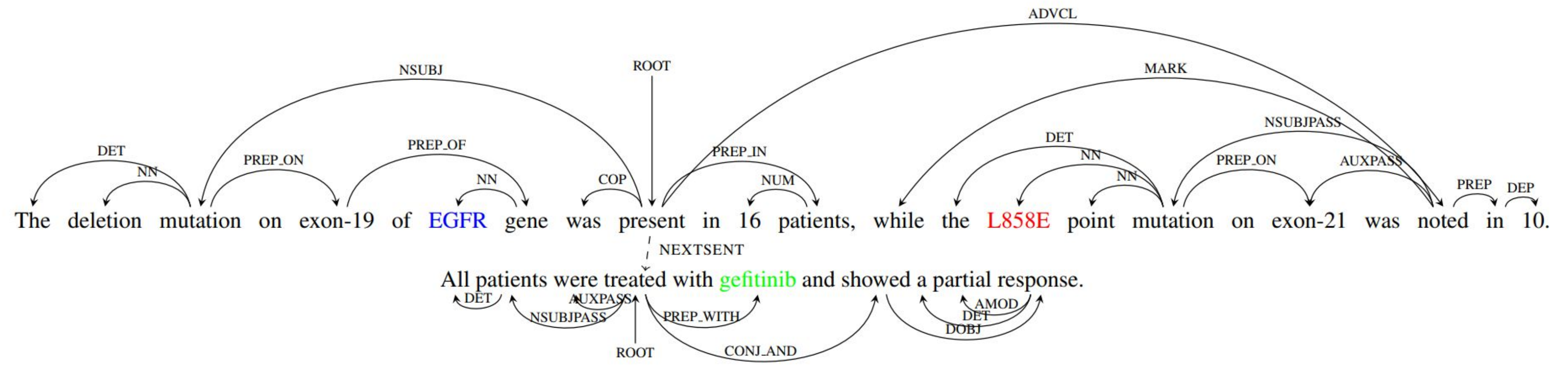


Figure 1: An example document graph for a pair of sentences expressing a ternary interaction (tumors with L858E mutation in EGFR gene respond to gefitinib treatment). For simplicity, we omit edges between adjacent words or representing discourse relations.

[“Cross-Sentence N-ary Relation Extraction with Graph LSTMs.” Peng, Poon, Quirk, Toutanova, and Yih (2017)]

Backpropagation with Document Graph

- If the graph has loops, then we are learning with loopy backpropagation: it might *fail* to converge!
- Partition graph into two groups:
  - Left-to-right edges + forward dependency arcs
  - Right-to-left edges + backward dependency arcs
- Creates DAGs for forward, backward passes

LSTM Gates

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
$$

https://en.wikipedia.org/wiki/Long_short-term_memory

- $\mathbf{f}_t$: Forget gate
- $\mathbf{i}_t$: Input gate
- $\mathbf{o}_t$: Output gate
- $\mathbf{c}_t$: Context embedding
- $\mathbf{h}_t$: Hidden state for token $\mathbf{t}$

- $\mathbf{x}_t$: embedding for token $\mathbf{t}$
- $\boldsymbol{\sigma}$: non-linear activation (e.g. ReLU)
- $\mathbf{W}_{\{i,o,c,f\}}$: learnable weight matrices on input $\mathbf{x}_t$
- $\mathbf{U}_{\{i,o,c,f\}}$: similar parameters , but for hidden states $\mathbf{h}_t$
- $\mathbf{b}_{\{i,o,c,f\}}$: bias term

Graph LSTM Gate Comparison

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i)$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o)$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c)$$
$$h_t = o_t \circ \sigma_h(c_t)$$

https://en.wikipedia.org/wiki/Long_short-term_memory

$$i_t = \sigma(W_i x_t + \sum_{j \in P(t)} U_i^{m(t,j)} h_j + b_i)$$
$$o_t = \sigma(W_o x_t + \sum_{j \in P(t)} U_o^{m(t,j)} h_j + b_o)$$
$$\tilde{c}_t = \tanh(W_c x_t + \sum_{j \in P(t)} U_c^{m(t,j)} h_j + b_c)$$
$$f_{tj} = \sigma(W_f x_t + U_f^{m(t,j)} h_j + b_f)$$
$$c_t = i_t \odot \tilde{c}_t + \sum_{j \in P(t)} f_{tj} \odot c_j$$
$$h_t = o_t \odot \tanh(c_t)$$

"Cross-Sentence N-ary Relation Extraction with Graph LSTMs."
Peng, Poon, Quirk, Toutanova, and Yih (2017)

Graph LSTM Gates

$$i_t = \sigma(W_i x_t + \sum_{j \in P(t)} U_i^{m(t,j)} h_j + b_i)$$

$$o_t = \sigma(W_o x_t + \sum_{j \in P(t)} U_o^{m(t,j)} h_j + b_o)$$

$$\tilde{c}_t = \tanh(W_c x_t + \sum_{j \in P(t)} U_c^{m(t,j)} h_j + b_c)$$

$$f_{tj} = \sigma(W_f x_t + U_f^{m(t,j)} h_j + b_f)$$

$$c_t = i_t \odot \tilde{c}_t + \sum_{j \in P(t)} f_{tj} \odot c_j$$

$$h_t = o_t \odot \tanh(c_t)$$

"Cross-Sentence N-ary Relation Extraction with Graph LSTMs."
Peng, Poon, Quirk, Toutanova, and Yih (2017)

- **`P(t)`**: predecessors of `t`; (dep. edges, adj. word)
- **`m(t,j)`**: connection type
- **$U_{\{c,o,i\}}^{m(t,j)}$**: edge type weight matrix
- Input i_t, output o_t gates depend on all predecessors
- Forget f_{tj} gate has single `j`

Results on Drug-Gene-Mutation Task

Model	Single-Sent.	Cross-Sent.
Feature-Based	74.7	77.7
CNN	77.5	78.1
BiLSTM	75.3	80.1
Graph LSTM - EMBED	76.5	80.6
Graph LSTM - FULL	**77.9**	**80.7**

Table 1: Average test accuracy in five-fold cross-validation for drug-gene-mutation ternary interactions. Feature-Based used the best performing model in (Quirk and Poon, 2017) with features derived from shortest paths between all entity pairs.

"Cross-Sentence N-ary Relation Extraction with Graph LSTMs." Peng, Poon, Quirk, Toutanova, and Yih (2017)

PSL for Taxonomy Induction Dive

What does this system do?

- Work in progress at Volley Labs
- Induces a concept taxonomy: subclass relationships
- Hearst patterns for extractors
- Equality rules that use WordNet synsets & cosine similarity of word embeddings (FB FastText vectors)
- Subsumption, superclass, etc. rules for structure
- PSL inference per document, then aggregate and infer for entire domain corpus (novel blocking technique)

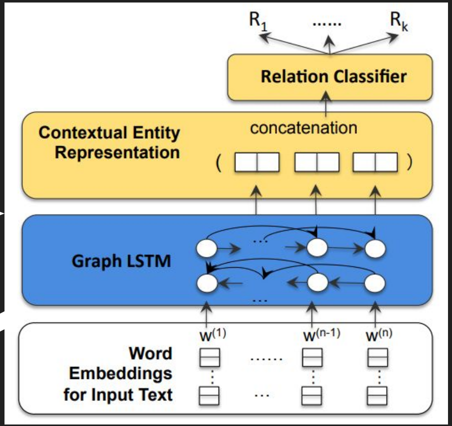
Examples and Results

- `mobile web payment < web payment` *(0.999887)*
- `web payment < payment` *(0.999974)*
- `collection mechanism < mechanism` *(0.992799)*
- `rate-based attack < attack` *(0.980801)*
- `server-side proxy < attack` *(0.967314)*
- Inferred ~ 4k subclass relationships from ~11k entities (with multiple mentions in text per entity!)

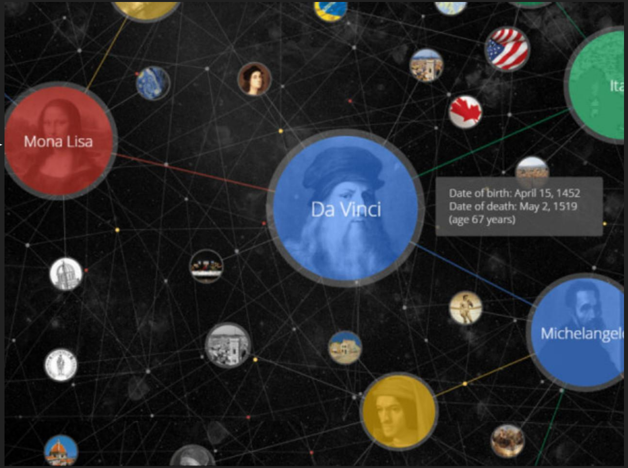
Combining the Best of Both Worlds: Deep Extractors & Collective Inference

Best of Both Worlds

- Learn the extractors!
  - Labeling functions, distant supervision first
  - Then learn low-parameter LSTMs
    - Fix input with pre-trained word embeddings
    - Up model complexity as labeled data increases
- Use probabilistic inference for collective reasoning and knowledge graph construction
  - PSL sorts thru labeling woes, inconsistent extractions

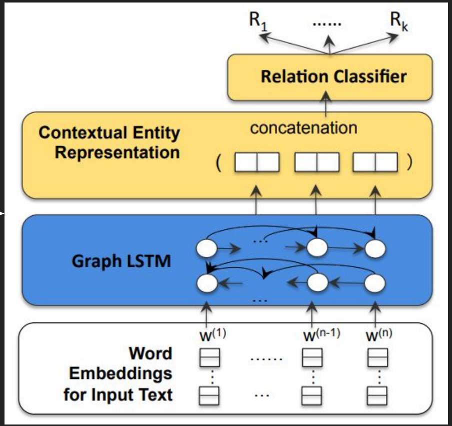


Probabilistic Soft Logic



1. https://i.pinimg.com/originals/8b/b2/04/8bb2047ba1cc290acb1256bc654acd59.jpg
2. https://upload.wikimedia.org/wikipedia/en/thumb/8/80/Wikipedia-logo-v2.svg/1122px-Wikipedia-logo-v2.svg.png
3. https://arxiv.org/pdf/1708.03743.pdf
4. https://psl.linqs.org/
5. https://devmesh.intel.com/projects/use-knowledge-graphs-to-do-anomaly-detection-job/activity

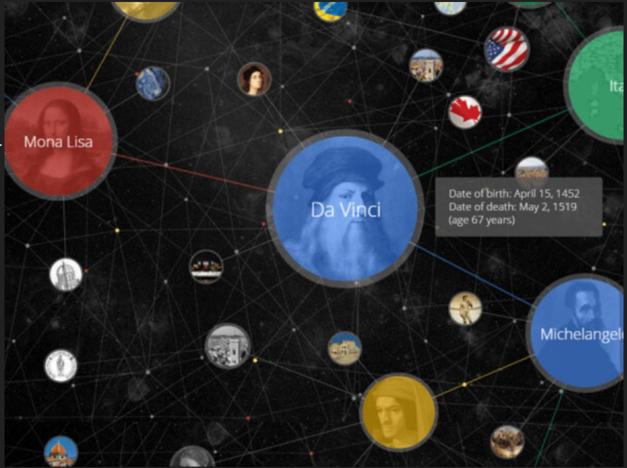
1. https://i.pinimg.com/originals/8b/b2/04/8bb2047ba1cc290acb1256bc654acd59.jpg
2. https://upload.wikimedia.org/wikipedia/en/thumb/8/80/Wikipedia-logo-v2.svg/1122px-Wikipedia-logo-v2.svg.png



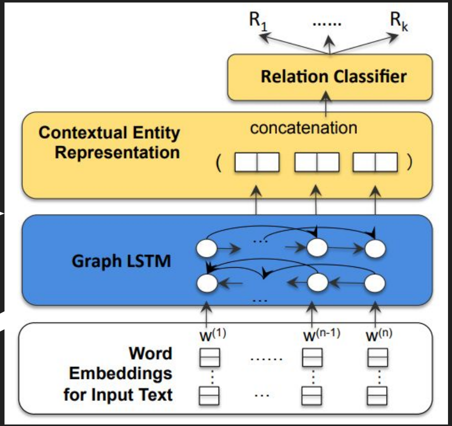
1. https://i.pinimg.com/originals/8b/b2/04/8bb2047ba1cc290acb1256bc654acd59.jpg
2. https://upload.wikimedia.org/wikipedia/en/thumb/8/80/Wikipedia-logo-v2.svg/1122px-Wikipedia-logo-v2.svg.png
3. https://arxiv.org/pdf/1708.03743.pdf

Probabilistic Soft Logic

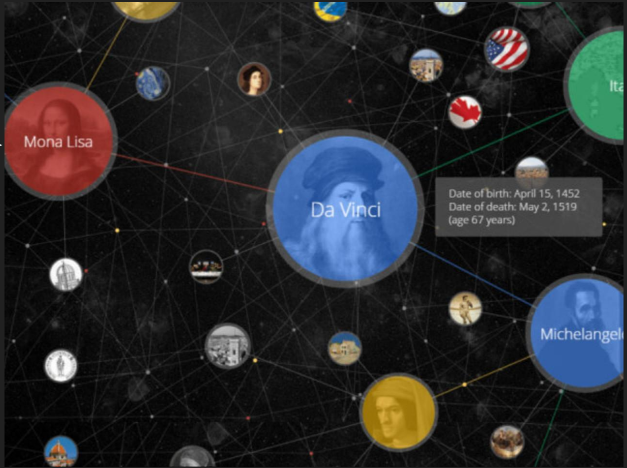
1. https://i.pinimg.com/originals/8b/b2/04/8bb2047ba1cc290acb1256bc654acd59.jpg
2. https://upload.wikimedia.org/wikipedia/en/thumb/8/80/Wikipedia-logo-v2.svg/1122px-Wikipedia-logo-v2.svg.png
3. https://arxiv.org/pdf/1708.03743.pdf
4. https://psl.linqs.org/



1. https://i.pinimg.com/originals/8b/b2/04/8bb2047ba1cc290acb1256bc654acd59.jpg
2. https://upload.wikimedia.org/wikipedia/en/thumb/8/80/Wikipedia-logo-v2.svg/1122px-Wikipedia-logo-v2.svg.png
3. https://arxiv.org/pdf/1708.03743.pdf
4. https://psl.linqs.org/
5. https://devmesh.intel.com/projects/use-knowledge-graphs-to-do-anomaly-detection-job/activity



Probabilistic
Soft
Logic



1. https://i.pinimg.com/originals/8b/b2/04/8bb2047ba1cc290acb1256bc654acd59.jpg
2. https://upload.wikimedia.org/wikipedia/en/thumb/8/80/Wikipedia-logo-v2.svg/1122px-Wikipedia-logo-v2.svg.png
3. https://arxiv.org/pdf/1708.03743.pdf
4. https://psl.linqs.org/
5. https://devmesh.intel.com/projects/use-knowledge-graphs-to-do-anomaly-detection-job/activity

In Conclusion

Outline Review

- Relation Primer
- Background on relation extraction literature
- Data labeling
- Shallow, multi-sentence RE w/ Probabilistic 1st order logic
- Deep Relation Extraction: Graph LSTMs
- Building a Knowledge Graph: PSL for Collective Inference
- Meta patterns in Relation Extraction

Takeaways

- Deep relation extraction outperforms shallow
- Multi-sentence systems have higher recall
- Semi-supervised / noisily labeled relation data is OK
- Probabilistic inference to make extractions consistent

Questions?

https://github.com/malcolmgreaves/talks/blob/master/deep_shallow_re_prob.pdf